IN THE CLAIMS

The current claims follow. For claims not marked as amended in this response, any difference in the claims below and the previous state of the claims is unintentional and in the nature of a typographical error.

1. (Currently Amended) An apparatus for executing at least one single program multiple data (SPMD) program, said apparatus comprising:

a micro single instruction multiple data (SIMD) unit associated with a microprocessor; and

a job buffer having an output coupled to an input of said micro SIMD unit,

wherein said job buffer is configured to:

at runtime, compare a job status of a plurality of jobs;

dynamically ~~bundles~~ bundle a subset of said plurality of jobs into a task based on on said comparison; ~~an equivalence of a job status of said jobs~~ and

~~allocates~~ allocate said task to said micro SIMD unit, and

wherein said job status ~~comprising~~ comprises a program counter value and a loop-counter list, and wherein a job is a combination of a program and an input data-set.

2. (Original) The apparatus as set forth in Claim 1 wherein said micro SIMD unit is capable of sending job status information to said job buffer.

3. (Original) The apparatus as set forth in Claim 1 wherein said at least one SPMD program comprises a plurality of input data streams having moderate diversification of control flows.

4. (Currently Amended) The apparatus as set forth in Claim 3 wherein said apparatus is configured to execute ~~executes~~ said at least one SPMD program once for each input data stream of said plurality of input data streams.

5. (Currently Amended) The apparatus as set forth in Claim 4 wherein said apparatus is configured to generate ~~generates~~ an instruction stream for each input data stream of said plurality of input data streams.

6. (Currently Amended) The apparatus as set forth in Claim 3 wherein said apparatus is configured to execute ~~executes~~ a plurality of SPMD programs and wherein each SPMD program of said plurality of SPMD programs is executed on a number of input data streams.

7. (Original) The apparatus as set forth in Claim 6 wherein said number of input data streams is greater than a program granularity threshold.

8. (Canceled).

9. (Currently Amended) The apparatus as set forth in Claim 1 wherein said apparatus is configured to perform ~~performs~~ job clustering to form a job bundle in which each job in said job bundle has an equivalent control flow.

10. (Currently Amended) The apparatus as set forth in Claim 9 wherein said apparatus is configured to perform ~~performs~~ said job clustering based on a job processing status of said jobs in said job bundle.

11. (Currently Amended) The apparatus as set forth in Claim 1 wherein said apparatus is configured to force ~~forces~~ a task to terminate at a point where a job control path might fork by placing a code-stop in said task.

12. (Currently Amended) The apparatus as set forth in Claim 11 wherein said apparatus is configured to minimize ~~minimizes~~ a required number of code-stops to be placed in said task by excluding from code-stop placement each control flow statements that is equivalent to a select instruction.

13. (Currently Amended) The apparatus as set forth in Claim 9 wherein said apparatus is configured to maximize ~~maximizes~~ a size of a job cluster by selecting tasks for execution in which a job processing status of each of said tasks is complete.

14. (Currently Amended) The apparatus as set forth in Claim 1 wherein said apparatus is configured to execute ~~executes~~ a data loading phase for a task before said apparatus executes a task execution phase for said task.

15. (Currently Amended) A method for executing at least one single program multiple data (SPMD) program, for use with a micro single instruction multiple data (SIMD) unit associated with a microprocessor, and a job buffer having an output coupled to an input of said micro SIMD unit, said method comprising the steps of:

~~providing a micro single instruction multiple data (SIMD) unit associated with a microprocessor;~~

~~providing a job buffer having an output coupled to an input of said micro SIMD unit; and~~

at runtime, comparing a job status of a plurality of jobs;

dynamically bundling a subset of said plurality of jobs into a task based on an equivalence of a job status of said jobs; and

allocating said task to said micro SIMD unit in said job buffer,

wherein said job status ~~comprising~~ comprises a program counter value and a loop-counter list, and wherein a job is a combination of a program and an input data-set.

16. (Original) The method as set forth in Claim 15 further comprising the step of: sending job status information from said SIMD unit to said job buffer.

17. (Original) The method as set forth in Claim 15 wherein said at least one SPMD program comprises a plurality of input data streams having moderate diversification of control flows.

18. (Original) The method as set forth in Claim 17 further comprising the step of:

executing said at least one SPMD program once for each input data stream of said plurality of input data streams.

19. (Original) The method as set forth in Claim 18 further comprising the step of:

generating an instruction stream for each input data stream of said plurality of input data streams.

20. (Original) The method as set forth in Claim 17 further comprising the steps of:

executing a plurality of SPMD programs; and

executing each SPMD program of said plurality of SPMD programs on a number of input data streams.

21. (Original) The method as set forth in Claim 20 wherein said number of input data streams is greater than a program granularity threshold.

22. (Canceled).

23. (Previously Presented) The method as set forth in Claim 15 further comprising the step of:

performing job clustering to form a job bundle in which each job in said job bundle has an equivalent control flow.

24. (Original) The method as set forth in Claim 23 further comprising the step of:

performing said job clustering based on a job processing status of said jobs in said job bundle.

25. (Previously Presented) The method as set forth in Claim 15 further comprising the step of:

forcing a task to terminate at a point where a job control path might fork by placing a code-stop in said task.

26. (Original) The method as set forth in Claim 25 further comprising the step of:

minimizing a required number of code-stops to be placed in said task by excluding from code-stop placement each control flow statements that is equivalent to a select instruction.

27. (Original) The method as set forth in Claim 23 further comprising the step of:

maximizing a size of a job cluster by selecting tasks for execution in which a job processing status of each of said tasks is complete.

28. (Previously Presented) The method as set forth in Claim 15 further comprising the step of:

executing a data loading phase for a task before executing a task execution phase for said task.